Current Efforts in MSF

1. Mission Simulation Dynamics Engine

An effort is under way to create a rover dynamics module called Mission Simulation Dynamics Engine (MSDE). The MSDE is based on the Open Dynamics Engine (ODE), a free software library for simulation of rigid body dynamics. The new module will serve as an alternative to the ROAMS rover kinematics/dynamics engine. The currently available MSF ROAMS module is limited to a single rover configuration and is not freely distributable. The new MSDE is designed to make the simulated rover fully user configurable through an XML description file and will be part of the MSF component library.

2. Alterative Run Time Interface

After the Defense Modeling and Simulation Office (DMSO) commercialized the Run-Time Infrastructure (RTI), the MSF has obtained RTI support through a license by Virtual Technology Corporation. The MSF team is evaluating another implementation of the RTI developed by a university. Experiments with the Georgia Tech RTI (FDK) have been performed and the basic MSF examples (with few disabled features) ran successfully. One feature that is of particular interest to the MSF is the ability to run the RTI using stared memory, because of its application to parallel processing on a super computer.

3. Simulation Launcher/Control Module

Other work in progress includes creating a graphical simulation launcher and controller to simplify the management of multiple federates. Because an MSF simulation can be run distributed on several machines, a graphical simulation launcher/controller will greatly simplify managing the details of running the simulation.

4. Simulation Definition

We are in the process of designing and implementing XML schemas, description files, and an XML parser to allow description of all participating components of a simulation, as well as initial parameters and simulation conditions. Rover definition is already defined and implemented to a large extent and current work is focused on extending the description scheme to other models such as power resources, instruments, environment, etc. The final simulation definition file will include other simulation parameters such as time step size,

failure probabilities, uncertainty conditions, and data to be collected.

5. Closer Integration with CLARAty

CLARAty is a software architecture for robotic autonomy and an emerging standard for interfacing with real and simulated robots. With the tight export restrictions removed, CLARAty is finally accessible to a larger user group. The MSF strives to be compatible with the CLARAty interface in order to give MSF users access to the library of robot control components for use in their simulation. Our initial focus is to change the MSF specific ROAMS interface to the CLARAty compliant interface. Other efforts include extending the MSF COM library with objects that match components in CLARAty.